САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Шилин Антон Алексеевич

Выпускная квалификационная работа бакалавра

Аппаратно-программный комплекс для дистанционного управления движением квадрокоптера

Направление 010400.62

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент Сотникова М. В.

Санкт-Петербург

2016

Содержание

Введение

Вопрос об управлении машинами и механизмами на расстоянии длительное время оставался нерешенным для человечества. Первое упоминание о реализации возможности дистанционного управления устройствами относится к 1898 году, когда Никола Тесла представил общественности радиоуправляемую лодку. Однако, меньше, чем через полвека в воздух поднялся первый беспилотный летательный аппарат, контролируемый пилотом с поверхности.

С тех пор появились новые способы дистанционной связи и расширился спектр воздушных аппаратов, для которых актуальна проблема управления на расстоянии. Одним из классов подобных устройств являются многороторные вертолеты, в частности, квадрокоптеры (рис. 1).

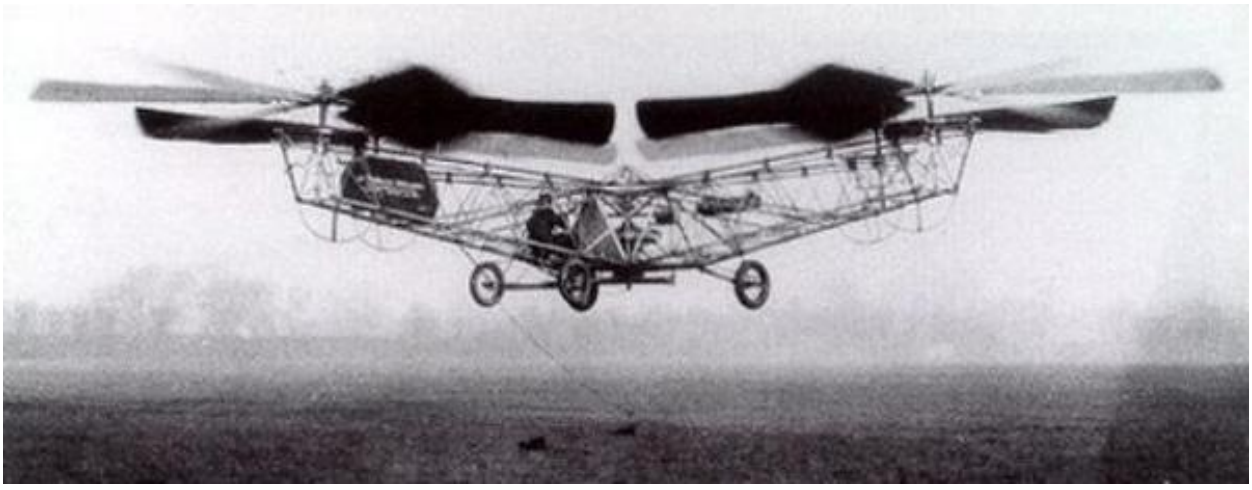


Рис. 1. Четырехроторный вертолет Ботезата, 1923.

В связи с ростом вычислительных мощностей современных микропроцессорных систем и увеличением требований к задачам, выполняемым летательными аппаратами, возникает потребность в автоматизации части работы оператора беспилотного устройства. Для решения этой проблемы необходимо создание автоматизированных комплексов для дистанционного управления беспилотными летательными аппаратами, которые реализуют на борту определенные алгоритмы

управления, а также предоставляют оператору возможность обзора телеметрии, поступающей с воздушного устройства. В зависимости от поставленных задач, у оператора должна быть возможность в выборе способов управления летательным аппаратом, например, задание определенного пути для следования.

В данной работе раскрывается вопрос создания аппаратно-программного комплекса для дистанционного управления движением четырехроторного вертолета (квадрокоптера). Актуальность обуславливается развитием сфер применения класса мультироторных беспилотных летательных аппаратов (БПЛА). Кроме того, большинство реализаций подобных комплексов, находящихся в открытом доступе, имеют весьма упрощенные алгоритмы управления и предназначены для использования в развлекательных целях.

Глава 1. Постановка задачи

1.1. Математическая модель квадрокоптера

Для описания динамики движения квадрокоптера введем две системы координат: инерциальную $OXYZ$, связанную с землей, и подвижную $O_B X_B Y_B Z_B$, связанную с рамой летательного аппарата. Обе системы являются правыми. Их схема представлена на рис. 2.

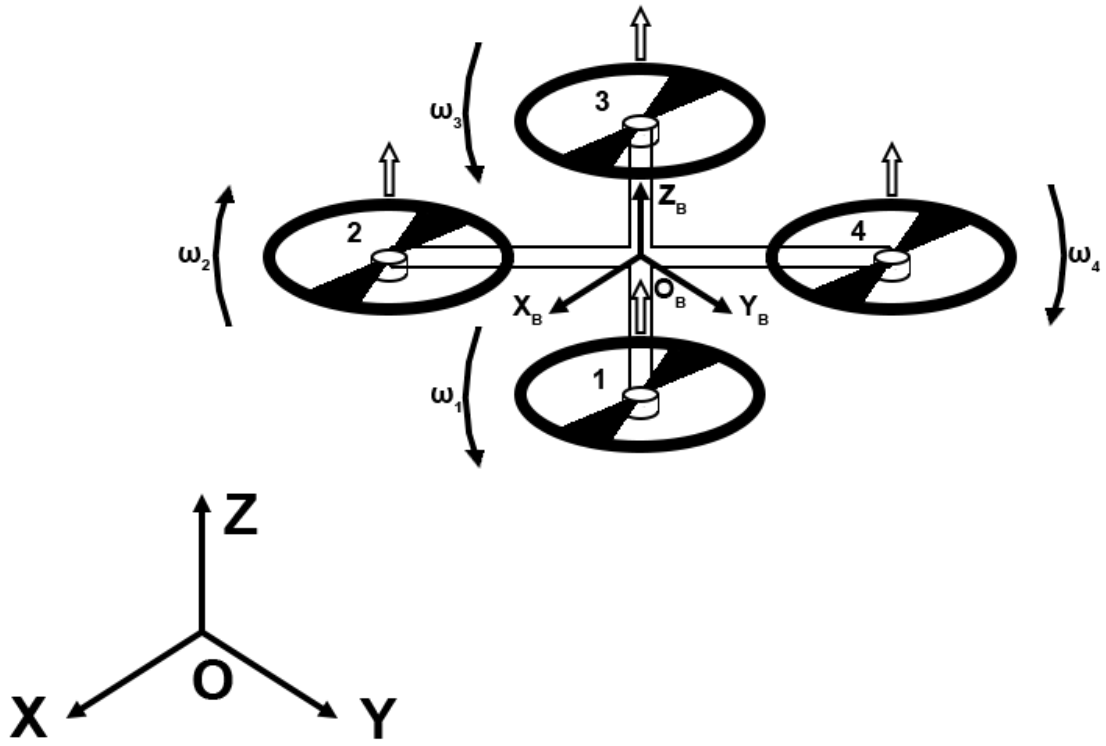


Рис. 2. Инерциальная и подвижная системы координат.

Способ задания осей инерциальной системы следующий: OX направлена на север, OY — на запад, а OZ — вверх, перпендикулярно земной поверхности.

Точка отсчета O_B подвижной системы находится в центре масс квадрокоптера. Будем считать, что он совпадает с точкой пересечения лучей рамы. Для подвижной системы оси зададим следующим образом: $O_B X_B$

направлена от точки пересечения лучей между первым и вторым моторами, $O_B Y_B$ — между первым и четвертым моторами, а $O_B Z_B$ — вверх, перпендикулярно плоскости лучей.

Положение центра масс квадрокоптера в инерциальной системе отсчета описывается вектором $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Ориентация подвижной системы координат относительно неподвижной определяется с помощью углов Эйлера $\boldsymbol{\eta}$:

$$\boldsymbol{\eta} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}.$$

Угол тангажа φ соответствует повороту квадрокоптера вокруг оси OY, угол крена θ — вокруг оси OX, а угол рысканья ψ — вокруг оси OZ.

В подвижной системе задаются вектор линейной скорости $\boldsymbol{V}$ и вектор угловой скорости $\boldsymbol{\nu}$:

$$\boldsymbol{V} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

Для перехода от подвижной системы к инерциальной используется следующая матрица преобразования координат $\boldsymbol{R}$:

$$\boldsymbol{R} = \begin{bmatrix} C_\psi C_\varphi & C_\psi S_\varphi S_\theta - S_\psi C_\theta & C_\psi S_\varphi C_\theta + S_\psi S_\theta \\ S_\psi C_\varphi & S_\psi S_\varphi S_\theta + C_\psi C_\theta & S_\psi S_\varphi C_\theta - C_\psi S_\theta \\ -S_\varphi & C_\varphi S_\theta & C_\varphi C_\theta \end{bmatrix},$$

где

$$S_x = \sin(x), C_x = \cos(x).$$

Связь между линейными скоростями $\dot{\boldsymbol{\xi}}$ и $\boldsymbol{V}$ в неподвижной и связанной системах координат задаются выражением:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{RV},$$

Аналогично, угловые скорости $\dot{\boldsymbol{\eta}}$ и $\boldsymbol{\nu}$ связаны соотношением:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{W}^{-1}\boldsymbol{\nu},$$

где матрица $\boldsymbol{W}^{-1}$ имеет вид [1]:

$$\boldsymbol{W}^{-1} = \begin{bmatrix} C_\psi / C_\theta & S_{\boldsymbol{\psi}} / C_\theta & 0 \\ -S_\psi & C_\psi & 0 \\ C_\psi T_\theta & S_\psi T_\theta & 1 \end{bmatrix},$$

$$T_x = \tan(x).$$

Будем считать квадрокоптер симметричным телом, у которого оси инерции совпадают с осями подвижной системы координат. В таком случае тензор инерции $\boldsymbol{I}$ примет следующий вид:

$$\boldsymbol{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}.$$

Введем следующие обозначения:

- m — масса аппарата.
- g — ускорение свободного падения.
- ω_i — угловая скорость вала мотора с индексом i.
- b — коэффициент воздушного сопротивления винта в плоскости вращения.
- k — коэффициент тяги винта.
- l — расстояние от оси вала мотора до центра масс квадрокоптера.
- T — суммарная тяга, создаваемая моторами летательного аппарата.
- $\tau_\varphi, \tau_\theta, \tau_\psi$ — моменты по соответствующим осям.

Управление квадрокоптером осуществляется путем задания определенных угловых скоростей валов моторов, что в свою очередь влияет на моменты и общую тягу:

$$\begin{gathered} \tau_\varphi = (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) lk \\ \tau_\theta = (\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) lk \\ \tau_\psi = (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) b \\ T = (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) k \end{gathered} \tag{1.1}$$

Справедливы следующие уравнения динамики:

$$\ddot{\boldsymbol{\eta}} = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\tau_\varphi}{I_y} \\ \frac{\tau_\theta}{I_x} \\ \frac{\tau_\psi}{I_z} \end{bmatrix},$$

$$\ddot{\boldsymbol{\xi}} = \frac{\boldsymbol{RT}}{m} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \frac{\boldsymbol{A}\dot{\boldsymbol{\xi}}}{m},$$

где $\boldsymbol{A}$ — матрица коэффициентов аэродинамического сопротивления, а $\boldsymbol{T}$ — вектор тяги моторов:

$$\boldsymbol{A} = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}.$$

Для упрощения математической модели будем считать, что углы φ и θ достаточно малы. Тогда уравнения динамики принимают итоговый вид:

$$\begin{cases} \dot{x} = C_\psi v_x - S_\psi v_y \\ \dot{y} = S_\psi v_x + C_\psi v_y \\ \dot{z} = v_z \\ \ddot{x} = C_\psi \varphi \frac{T}{m} + S_\psi \theta \frac{T}{m} - \frac{A_x}{m}\dot{x} \\ \ddot{y} = S_\psi \varphi \frac{T}{m} - C_\psi \theta \frac{T}{m} - \frac{A_y}{m}\dot{y} \\ \ddot{z} = \frac{T}{m} - g - \frac{A_z}{m}\dot{z} \\ \dot{\varphi} = C_\psi p + S_\psi q \\ \dot{\theta} = -S_\psi p + C_\psi q \\ \dot{\psi} = r \\ \ddot{\varphi} = \frac{C_\psi \tau_\varphi}{I_y} + \frac{S_\psi \tau_\theta}{I_x} \\ \ddot{\theta} = -\frac{S_\psi \tau_\varphi}{I_y} + \frac{C_\psi \tau_\theta}{I_x} \\ \ddot{\psi} = \frac{\tau_\psi}{I_z} \end{cases} \quad (1.2)$$

Далее под "положением летательного аппарата в пространстве" будет подразумеваться положение центра масс квадрокоптера в инерциальной системе отсчета.

1.2. Формулировка задачи

Разработать комплекс для дистанционного управления движением квадрокоптера, который должен:

- предоставлять возможность управления квадрокоптером при помощи мобильного устройства (смартфона/планшета) на дистанциях до 50 м от местонахождения пульта управления;
- осуществлять непрерывную синхронизацию мобильного устройства и бортовой части (квадрокоптера) по каналу Wi-Fi;
- предоставлять пользователю полный доступ к получаемой с датчиков летательного аппарата телеметрии;
- предоставлять пользователю интуитивно-понятный интерфейс для управления квадрокоптером;
- содержать на борту БПЛА алгоритмы управления для приведения квадрокоптера в определенную точку пространства с заданным углом рысканья;
- содержать на борту БПЛА астатический регулятор по положению в пространстве для компенсации внешнего воздействия (ветер).

Пусть $\boldsymbol{\xi} = (x, y, z)$ — координаты центра масс летательного аппарата в неподвижной инерциальной системе отсчета, $\boldsymbol{\eta} = (\varphi, \theta, \psi)$ — ориентация БПЛА в этой системе; и имеется набор измерений $\boldsymbol{Y}$.

Под задачей стабилизации квадрокоптера с определенными углами ориентации $\boldsymbol{\eta_d}$ будем подразумевать поиск такого управления $u = u(\boldsymbol{Y}, \boldsymbol{\eta_d})$, чтобы $\boldsymbol{\eta} \to \boldsymbol{\eta}_d$ при $t \to \infty$.

Пусть задана некоторая точка $P_d(\boldsymbol{\xi}_d)$ и угол ψ_d. В таком случае задача приведения квадрокоптера в определенную точку пространства с заданным углом рысканья состоит в следующем: найти такое управление $u = u(\boldsymbol{Y}, P_d)$, чтобы $\boldsymbol{\xi} \to \boldsymbol{\xi}_d$ и $\psi \to \psi_d$ при $t \to \infty$.

Пусть $P_s(\boldsymbol{\xi}_s)$ — координаты точки, в которой необходимо стабилизировать летательный аппарат. Под астатическим регулятором

подразумевается такое управление $u = u(\boldsymbol{Y}, P_s)$, что при наличии постоянного внешнего воздействия H на положение центра масс квадрокоптера, выполняется $\boldsymbol{\xi} \to \boldsymbol{\xi}_s$ при $t \to \infty$.

Исходя из изложенных требований были поставлены следующие задачи:

- разработать логику и структуру работы аппаратно-программного комплекса для дистанционного управления движением квадрокоптера.
- в рамках комплекса разработать бортовые алгоритмы управления для стабилизации летательного аппарата с определенными углами ориентации; для приведения летательного аппарата в заданную точку пространства с заданным углом рысканья; для компенсации постоянного внешнего воздействия на положение центра масс летательного аппарата в пространстве. Произвести имитационное моделирование динамики движения летательного аппарата с применением разработанных законов управления.

В настоящее время уже ведутся разработки комплексов для управления движением мультикоптеров при помощи мобильных устройств, однако, большинство из них носит любительский характер.

При рассмотрении отдельно бортовой части стоит отметить, что существуют готовые коммерческие решения плат управления полетом [2][3]. Среди реализованных возможностей представлены алгоритмы стабилизации по углам, а также стабилизация в точке на основе данных датчиков геолокации. Недостатком подобных решений является высокая стоимость и низкая точность определения положения в пространстве, ошибка которых достигает десятков метров.

В свободном доступе представлено несколько алгоритмов стабилизации квадрокоптера по углам [4]. Также присутствуют методы

приведения летательного аппарата в определенную точку пространства [5], но они не учитывают внешнего возмущения. Таким образом, необходимо осуществить разработку астатического регулятора по положению в пространстве для компенсации воздействия на летательный аппарат внешних сил таких, как ветер.

Глава 2. Разработка аппаратно-программного комплекса

2.1. Аппаратная архитектура комплекса

Комплекс можно условно разделить на две крупные части: базовую станцию, которая включает в себя пульт управления и коммуникационный блок, и бортовую часть.

Рассмотрим подробнее каждую составляющую.

Бортовая часть

Компоненты бортовой части (рис. 3) можно разделить на 4 блока:

- блок датчиков;
- блок коммуникации;
- блок исполнительных устройств;
- вычислительный блок.

Рис. 3. Структура бортовой части.

Блок датчиков содержит следующие компоненты:

- трехосевой акселерометр;
- трехосевой гироскоп;
- барометр (датчик атмосферного давления);
- ультразвуковой датчик расстояния (сонар);
- датчик температуры окружающего воздуха;
- вольтметр;
- 4 инфракрасных датчика обнаружения препятствий.

Бортовой акселерометр возвращает оценку значений ускорений $\widehat{v_x}, \widehat{v_y}, \widehat{v_z}$ в связанной системе координат. При интегрировании этих данных находятся соответствующие оценки скоростей $\widehat{v_x}, \widehat{v_y}, \widehat{v_z}$.

Гироскоп возвращает оценку угловых скоростей $\hat{p}, \hat{q}, \hat{r}$.

Ультразвуковой датчик расстояния используется для нахождения высоты над поверхностью. Сонар находится на нижней стороне летательного аппарата и направлен вдоль отрицательного направления оси $O_B Z_B$. В качестве возвращаемого значения выступает время t_s прохождения звуковым импульсом расстояния до ближайшего препятствия, находящегося в секторе работы датчика, и обратно. Данный сенсор имеет ограниченный рабочий диапазон. При удалении от препятствия дальше верхней допустимой границы значительно увеличивается погрешность получаемых данных. Оценка расстояния до поверхности осуществляется следующим образом:

$$\hat{h} = \frac{t_s v_s}{2},$$

где v_s — скорость звука в воздухе.

Точность показаний подлежит увеличению путем учета текущей температуры воздуха при определении v_s. Для этого используются данные T_a температурного датчика (значение в Кельвинах). Вычисления осуществляются по следующей формуле [6]:

$$v_s = \sqrt{\frac{\gamma R T_a}{M}},$$

где γ — показатель адиабаты, R — универсальная газовая постоянная, M — молярная масса воздуха.

Барометр предоставляет данные о текущем атмосферном давлении ρ на уровне квадрокоптера, которые позволяют произвести оценку высоты $\hat{h}$ летательного аппарата в том случае, если расстояние до поверхности находится за рамками рабочего диапазона ультразвукового датчика. Для этого используется следующая формула [7]:

$$\hat{h} = 44330\left(1 - \frac{\rho}{\rho_0}^{\frac{1}{5.255}}\right).$$

Если возвышение над поверхностью позволяет задействовать ультразвуковой сенсор, то происходит калибровка датчика атмосферного давления, при которой его текущее возвращаемое значение ρ_0 запоминается и соотносится с данными сонара, что позволяет увеличить точность оценки высоты.

Вольтметр осуществляет измерение текущего уровня заряда аккумуляторных батарей летательного аппарата.

Инфракрасные датчики обнаружения препятствий направлены в плоскости $O_B X_B Y_B$ по положительным и отрицательным направлениям осей $O_B X_B$ и $O_B Y_B$. В качестве возвращаемого значения для каждого из датчиков выступает 1 при нахождении препятствия на оси действия датчика ближе определенной дистанции и 0 при его отсутствии.

Блок коммуникации состоит из беспроводного интерфейса. Данный модуль организует точку доступа Wi-Fi и осуществляет прием и отправку данных на базовую станцию.

Блок исполнительных устройств включает в себя 4 электронных регулятора хода (ESC) и 4 бесколлекторных двигателя. Регуляторы ESC служат для плавного варьирования оборотов электродвигателей летательного

аппарата.

Вычислительный блок представляет собой плату микроконтроллера, на котором запускается главный цикл рабочей программы летательного аппарата, выполняющий обработку поступающей информации и синтезирующий управляющий сигнал для исполнительных устройств.

Базовая станция

В качестве пульта управления выступает программное приложение, реализованное на мобильном устройстве, имеющем сенсорный экран для ввода информации, с модулем беспроводной связи стандарта IEEE 802.11n Wi-Fi. В качестве предустановленной операционной системы была выбрана Android OS из-за обширной базы мобильных устройств, работающих на ее основе. Язык разработки является Java 8. Такой выбор был произведен в первую очередь из-за специализации Android-устройств для работы с приложениями, запущенными на Java Virtual Machine.

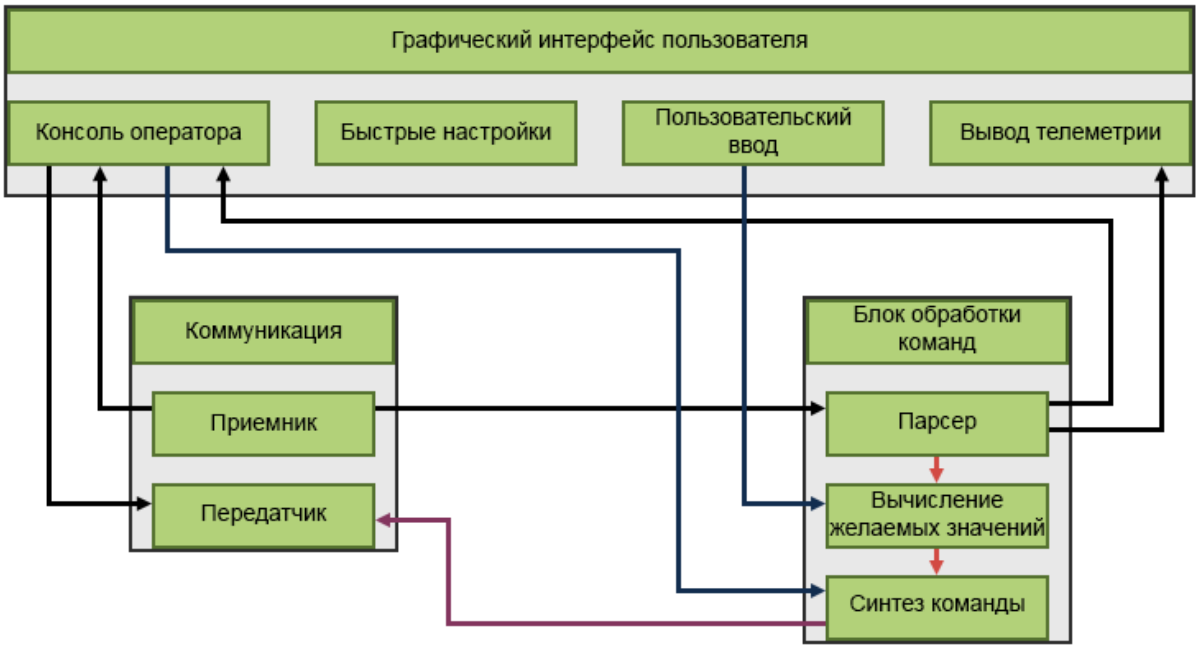


Рис. 4. Структура базовой станции.

Базовая станция (рис. 4), на которой установлено программное приложение комплекса, включает в себя три блока:

- блок графического интерфейса пользователя (GUI);
- блок коммуникации;
- блок обработки команд.

Пользователь вводит команды с помощью графической реализации пульта управления типа "джойстик". В распоряжении оператора имеется два рычага управления: левый отвечает за вертикальное перемещение летательного аппарата по оси OZ и за поворот вокруг этой оси (изменение угла рысканья); правый отвечает за перемещение в горизонтальной плоскости OXY. На уровне приложения введенные данные могут интерпретироваться в двух режимах:

1. режим задания определенных углов ориентации;
2. режим преследования мнимой точки.

При работе в первом режиме приложение в качестве управляющей команды, заданной пользователем, передает бортовой части желаемые углы ориентации $\varphi_d, \theta_d, \psi_d$ и желаемую вертикальную скорость $\dot{z_d}$. При переключении во второй режим приложение высчитывает координаты мнимой точки $D(x_d, y_d, z_d)$ (в подвижной системе координат). Под мнимой точкой подразумевается точка в пространстве, в которую необходимо привести центра масс квадрокоптера. В данном случае управляющая команда будет компоноваться на основе желаемого угла рысканья ψ_d и координат x_d, y_d, z_d.

2.2. Программная архитектура комплекса

Как и в случае аппаратной архитектуры, программная составляющая делится на две основные части: бортовое программное обеспечение и приложение базовой станции.

Бортовая часть

Основная программа летательного аппарата выполняется в ходе

срабатывания специального таймера, который имеет определенную частоту, зависящую, в свою очередь, от частоты работы бортового микроконтроллера.

Программа бортовой части должна выполнять следующие функции:

- проводить инициализацию блока исполнительных устройств при запуске летательного аппарата;
- обеспечивать постоянное взаимодействие с периферийными модулями такими, как датчики, беспроводной интерфейс;
- осуществлять непрерывное дистанционное соединение и синхронизацию с пультом управления базовой станции;
- поддерживать алгоритмы для аварийных ситуаций (потеря сигнала пульта управления, сильный разряд бортовых батарей, избежание столкновения с боковыми препятствиями на низких скоростях, необходимость экстренного отключения двигателей);
- при необходимости осуществлять фильтрацию информации, получаемой с периферийных датчиков;
- реализовывать алгоритмы управления, рассмотрение которых происходит в параграфе 2.3;
- предоставлять возможность настройки подключенного оборудования;

Далее рассмотрим программные решения для выполнения описанных выше функций.

Инициализация блока исполнительных устройств при запуске летательного аппарата. При старте программы необходимо выполнить набор низкоуровневых команд для инициализации электронных регуляторов хода. Это необходимо для передачи информации о нижней и верхней границах последующих управляющих сигналов.

Взаимодействие с периферийными модулями. Как правило, производители оборудования внешних компонентов предлагают вместе с устройством библиотеки драйверов для работы с ними. Таким образом, для

получения информации с соответствующего датчика или отправки данных через беспроводной интерфейс достаточно воспользоваться соответствующими функциями из прилагаемых библиотек. Обычно опрос внешних сенсоров происходит в параллельном потоке выполнения программы, который имеет меньшую частоту таймера. Это позволяет эффективнее использовать ресурсы микроконтроллера, не нагружая его дополнительными вычислениями.

Непрерывное дистанционное соединение и синхронизация с пультом управления базовой станции осуществляется при помощи беспроводного интерфейса. Данный модуль позволяет создавать новую беспроводную сеть и выполняет роль точки доступа и сервера. Для создания канала связи между базовой станцией и летательным аппаратом необходимо:

1. начать работу основной программы бортовой части;
2. подождать запуска беспроводной точки доступа;
3. выполнить подключение базовой станции по Wi-Fi каналу к точке доступа в режиме клиента.

Далее запускается дополнительный таймер в отдельном потоке, предназначенный для опроса беспроводного интерфейса. В ходе работы данной подпрограммы происходит считывание полученных от пульта управления командных сигналов из стека модуля и отправка накопленных данных телеметрии.

В случае отсутствия команд от оператора посылается тривиальная команда для поддержки связи.

После получения командных сигналов, необходимо произвести их интерпретацию. Структура полученного сообщения выглядит следующим образом:

$$U = n\_u_{num}\_args,$$

где n — двухзначная длина сигнала в байтах , u_{num} — трехзначный номер команды из списка интерпретатора, $args$ — список аргументов командного

сигнала. В качестве примера рассмотрим команду на изменение угловой скорости первого мотора:

$$U = 11\_102\_1\_70$$

В приведенном сигнале $n = 11$, $u_{num} = 102$, $args = 1\_70$, где 1 — номер соответствующего электродвигателя, 70 — угловая скорость мотора в процентном соотношении от максимальной. Интерпретатор производит расшифровку полученного сигнала и передает необходимые команды в другие блоки программы.

Алгоритмы для аварийных ситуаций.

После получения команд с пульта управления производится анализ возможности их выполнения, который необходим в целях безопасности. Одним из проявлений работы данной функции выступает отказ движения в сторону препятствия, если оно в данный момент зафиксировано боковым инфракрасным датчиком. При генерации управления осуществляется схожая проверка, которая также включает в себя контроль безопасности по высоте. Он заключается в модификации желаемой координаты z_d и выглядит следующим образом:

$$z_d = \begin{cases} z_c - h, z_d < z_c - h \\ z_d, z_d > z_c - h \end{cases}$$

В этом выражении z_c — текущее положение по высоте в инерциальной системе отсчета, h — высота над поверхностью.

В аварийной ситуации, вызванной потерей сигнала с пульта управления, подается автоматическая команда на посадку летательного аппарата:

$$x_d = x_c$$
$$y_d = y_c$$
$$z_d = 0$$

где x_c и y_c — положение квадрокоптера в плоскости OXY в момент потери связи, x_d и y_d — желаемые координаты. Особенностью этого способа решения внештатной ситуации является предположение, что под

летательным аппаратом находится земная поверхность, на которую возможна посадка.

Схожим образом реализован алгоритм действий в ситуации сильного разряда бортовых батарей. Отличием выступает возможность оператора управлять движением квадрокоптера в горизонтальной плоскости OXY (для пользователя блокируется управление по высоте):

$$z_d = 0.$$

Необходимость экстренного отключения электродвигателей возникает при невозможности избежать столкновения с препятствием, находящимся на пути движения летательного аппарата, что призвано в первую очередь обезопасить живые существа, находящиеся поблизости от квадрокоптера. Данный механизм защиты может быть активирован либо по команде оператора, либо при обнаружении препятствия инфракрасными датчиками или ультразвуковым датчиком расстояния в опасной окрестности с условием, что скорость квадрокоптера превышает определенное значение.

Фильтрация информации, получаемой с периферийных датчиков. Как правило, производители MEMS датчиков предоставляют встроенный функционал для фильтрации получаемых данных в своих устройствах такие, как комплементарный фильтр, DCM фильтр и другие.

В настоящей работе рассматривается применение DCM фильтра [8] для оценки углов ориентации летательного аппарата. Блок-схема алгоритма его работы представлена на рис. 5. Как видно из схемы, фильтр может использовать показания магнитометра, который зачастую входит в состав инерциальных измерительных устройств. Достоинством данного метода оценки ориентации является компенсация дрейфа гироскопа с течением времени в сравнении с обычным комплементарным фильтром.

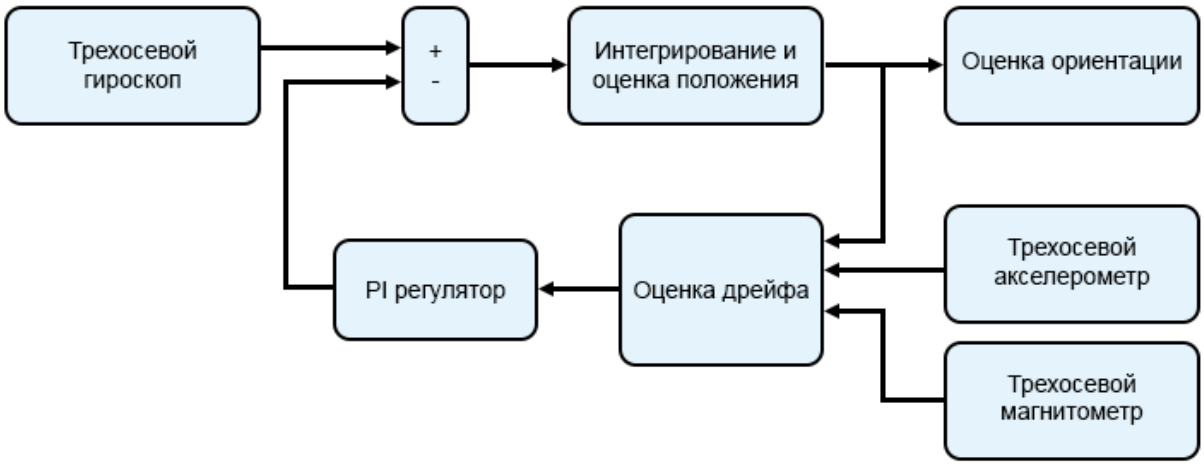


Рис. 5. Алгоритм работы DCM фильтра.

Оценка текущих скоростей в подвижной системе координат производится интегрированием ускорений, полученных с помощью фильтрации данных акселерометра. Последнее необходимо для устранения шумов, присутствующих в показаниях датчика. Для выполнения этой задачи, проектируемый комплекс содержит модифицированный фильтр низких частот. Его работа описывается следующей формулой:

$$\widehat{R_n} = \begin{cases} \widetilde{R_n}, \left(\widetilde{R_n} - \widetilde{R_{n-1}}\right) > \varepsilon \\ \widehat{R_{n-1}} + \alpha\left(\widetilde{R_n} - \widehat{R_{n-1}}\right), \left(\widetilde{R_n} - \widetilde{R_{n-1}}\right) \leq \varepsilon \end{cases},$$

где $\hat{R}$ — фильтрованная оценка по одной оси, $\tilde{R}$ — нефильтрованные данные сенсора, $\alpha \in [0; 1]$ — коэффициент фильтрации, ε — пороговое значение для фильтрации.

После фильтрации и предварительной обработки данных с периферийных датчиков под "компонентами вектора состояния" подразумеваются их оценки.

Алгоритмы управления. Их программная реализация осуществляется в теле основного таймера, имеющего высший приоритет выполнения. Для данного блока программы предоставляются командные сигналы, передаваемые оператором, и информация с фильтров периферийных данных. Подробнее описание законов управления будет приведено в следующей

главе.

Настройка подключенного оборудования. Некоторые периферийные датчики и электронные регуляторы хода предоставляют возможность программной настройки. Для удобства оператора в комплексе разрабатывается возможность удаленной отладки оборудования летательного аппарата при помощи консоли пульта управления.

Базовая станция

Как было упомянуто выше, пульт управления базовой станции комплекса представляет собой мобильное приложение. Рассмотрим выполняемые им функции:

- непрерывное дистанционное соединение и синхронизация с бортовой частью комплекса после подключения;
- предоставление оператору полной информации о получаемой телеметрии;
- поддержка двух вариантов управления: режим задания определенных углов ориентации и режим преследования мнимой точки;
- реализация команды на аварийное выключение электромоторов летательного аппарата;
- консоль оператора для настройки комплекса.

В целом алгоритм передачи данных имеет такую же структуру, как и бортовой блок коммуникации. Для начала работы комплекса необходимо, чтобы базовая станция была подключена к точке доступа бортового беспроводного интерфейса. После команды оператора на соединение приложение запускает блок подключения в режиме клиента в отдельном потоке-таймере. Частота данного таймера совпадает с аналогом, задействованным на борту летательного аппарата. Дальнейшее описание принципа работы интерпретатора не имеет смысла, так как он имеет

незначительные отличия от бортовой реализации.

Вывод полученной телеметрии осуществляется на экран управления летательным аппаратом. Для значений текущих углов ориентации φ, θ, ψ, линейных скоростей $\dot{x}, \dot{y}, \dot{z}$ в неподвижной системе координат и высоты квадрокоптера h относительно поверхности информация отображается в численных значениях. Кроме того, оператор получает доступ к информации о текущем уровне заряда бортовых батарей.

В окне управления присутствует схематичная модель квадрокоптера, отображающая текущую ориентацию в инерциальной системе координат. При нажатии на эту область экрана происходит переключение между режимами управления. Отдельные кнопки служат для подключения (отключения) к летательному аппарату, вызова команды на аварийную остановку двигателей.

Основные элементы на экране управления — это графическая реализация пульта управления типа "джойстик". Подобное интуитивно понятное решение призвано облегчить процесс управления квадрокоптером. При подаче команды оператором (например, изменение положение рычага управления) начинается компоновка командного сигнала. Она производится в полном соответствии со списком команд интерпретатора. Далее командный сигнал помещается в стек данных для отправки коммуникационного блока.

Комплекс содержит консоль оператора в отдельном окне, которая позволяет осуществлять настройку оборудования и отправку информационных пакетов путем текстовых команд. Для удобства оператора разработана система справки для каждой команды.

2.3. Разработка законов управления

Исходя из задачи приведения квадрокоптера в определенную точку пространства с заданным углом рысканья, необходимо разработать такой

закон управления, который переводил бы летательный аппарат из текущего положения в пространстве $P_c(x_c, y_c, z_c)$ с углом рысканья ψ_c в желаемое положение $P_d(x_d, y_d, z_d)$ с углом рысканья ψ_d.

Математическая модель (1.2), полученная ранее, является нелинейной, что осложняет ее анализ. С другой стороны, это позволяет осуществлять построение закона управления рысканьем.

Для синтеза закона управления используются следующие данные, полученные после фильтрации и предварительных вычислений:

- скорости $\dot{x}, \dot{y}, \dot{z}$;
- текущее положение x, y, z;
- угловые скорости $\dot{\varphi}, \dot{\theta}, \dot{\psi}$;
- углы ориентации φ, θ, ψ.

Управляющий сигнал для электромоторов подлежит ограничению из-за технических особенностей летательного аппарата. Угловая скорость для каждого исполнительного устройства лежит в диапазоне $[0; 10000]$ об/мин.

Стабилизация по углам наклона

Для стабилизации по углам наклона и высоте летательного аппарата используются методы ПИД регулирования. Предполагается, что компоненты тензора инерции $\boldsymbol{I}$ квадрокоптера приблизительно известны, тогда управляющие моменты по соответствующим углам вычисляются по следующим формулам:

$$\begin{aligned} \tau_\varphi &= \left(k_{\varphi,d}(\dot{\varphi}_d - \dot{\varphi}) + k_{\varphi,p}(\varphi_d - \varphi)\right) I_y \\ \tau_\theta &= \left(k_{\theta,d}(\dot{\theta}_d - \dot{\theta}) + k_{\theta,p}(\theta_d - \theta)\right) I_x \\ \tau_\psi &= \left(k_{\psi,d}(\dot{\psi}_d - \dot{\psi}) + k_{\psi,p}(\psi_d - \psi)\right) I_z \end{aligned}, \qquad (2.3.1)$$

где $k_{\varphi,d}$, $k_{\varphi,p}$, $k_{\theta,d}$, $k_{\theta,p}$, $k_{\psi,d}$, $k_{\psi,p}$ — постоянные вещественные числа, выбранные с учетом устойчивости положения равновесия замкнутой системы (1.2), (2.3.1).

В данном законе подразумевается, что общая тяга должна компенсировать силу притяжения:

$$T = mg.$$

Из уравнений (1.1) выводятся следующие выражения:

$$\begin{aligned}
\omega_1 &= \sqrt{\frac{T}{4k} - \frac{\tau_\varphi}{4kl} + \frac{\tau_\theta}{4kl} + \frac{\tau_\psi}{4b}} \\
\omega_2 &= \sqrt{\frac{T}{4k} - \frac{\tau_\varphi}{4kl} - \frac{\tau_\theta}{4kl} - \frac{\tau_\psi}{4b}} \\
\omega_3 &= \sqrt{\frac{T}{4k} + \frac{\tau_\varphi}{4kl} - \frac{\tau_\theta}{4kl} + \frac{\tau_\psi}{4b}} \\
\omega_4 &= \sqrt{\frac{T}{4k} + \frac{\tau_\varphi}{4kl} + \frac{\tau_\theta}{4kl} - \frac{\tau_\psi}{4b}}
\end{aligned} \qquad (2.3.2)$$

Подставив вычисленные в (2.3.1) значения в (2.3.2), находятся необходимые угловые скорости для электромоторов летательного аппарата.

Исходя из предположений математической модели (1.2) были установлены следующие ограничения: $\varphi_d \in [-0.5; 0.5]$, $\theta_d \in [-0.5; 0.5]$ (величины указаны в радианах).

Достижение заданной точки пространства

Закон управления для достижения квадрокоптером заданной точки $P_d(x_d, y_d, z_d)$ с углом рысканья ψ_d из текущего положения $P_c(x_c, y_c, z_c)$ с углом рысканья ψ_c базируется на следующем алгоритме:

1. вычисление желаемых ускорений $\tilde{\ddot{x}}, \tilde{\ddot{y}}, \tilde{\ddot{z}}$;
2. вычисление желаемых углов ориентации $\tilde{\varphi}, \tilde{\theta}, \tilde{\psi}$ и общей тяги $\tilde{T}$;
3. использование закона стабилизации по углам наклона.

Рассмотрим подробнее каждый шаг.

Вычисление желаемых ускорений. На данном этапе происходит генерация таких ускорений $\tilde{\ddot{x}}, \tilde{\ddot{y}}, \tilde{\ddot{z}}$, которые обеспечивают достижение необходимого положения P_d в каждый момент времени t. Для их вычислений также применяется пропорционально-дифференциальный регулятор:

$$\tilde{\ddot{x}} = \left(k_{x,d}(\dot{x_d} - \dot{x}) + k_{x,p}(x_d - x)\right)$$
$$\tilde{\ddot{y}} = \left(k_{y,d}(\dot{y_d} - \dot{y}) + k_{y,p}(y_d - y)\right) \quad (2.3.3)$$
$$\tilde{\ddot{z}} = \left(k_{z,d}(\dot{z_d} - \dot{z}) + k_{z,p}(z_d - z)\right)$$

где $k_{x,d}$, $k_{x,p}$, $k_{y,d}$, $k_{y,p}$, $k_{z,d}$, $k_{z,p}$ — постоянные вещественные числа, выбранные с учетом устойчивости положения равновесия замкнутой системы (1.2), (2.3.3).

Для ускорения достижения конечной точки и уменьшения перерегулирования (снижения времени на стабилизацию в заданном положении) приведенный метод нуждается в доработке из-за особенностей динамики летательного аппарата.

Вычисление желаемых углов ориентации и общей тяги происходит на том условии, что при достижении летательным аппаратом полученных значений, будут обеспечены желаемые ускорения в данный момент времени.

Из математической модели (1.2) следует:

$$\tilde{\varphi} = \frac{\left(C_{\psi_d}\tilde{\ddot{x}} + S_{\psi_d}\tilde{\ddot{y}}\right)}{g}$$
$$\tilde{\theta} = \frac{\left(S_{\psi_d}\tilde{\ddot{x}} - C_{\psi_d}\tilde{\ddot{y}}\right)}{g}$$
$$\tilde{T} = m\left(\tilde{\ddot{z}} + g\right)$$

Использование закона стабилизации по углам наклона выполняется для достижения найденных углов тангажа $\tilde{\varphi}$ и крена $\tilde{\theta}$, а также конечного угла рысканья ψ_d.

Разработанный алгоритм является решением задачи приведения квадрокоптера в заданную точку с определенным углом рысканья в каждый момент времени t.

Астатический регулятор по положению в пространстве

Для решения задачи подавления постоянного внешнего воздействия на

летательный аппарат такого, как ветер, была произведена модификация предыдущего закона управления.

Уменьшение отклонения от заданного положения в пространстве достигается путем добавления в регулятор (2.3.3), отвечающий за вычисление желаемых ускорений, интегральной составляющей:

$$\begin{aligned}\tilde{\ddot{x}} &= \left(k_{x,d}(\dot{x_d} - \dot{x}) + k_{x,p}(x_d - x) + k_{x,i}\int(x_d - x)\right)\\ \tilde{\ddot{y}} &= \left(k_{y,d}(\dot{y_d} - \dot{y}) + k_{y,p}(y_d - y) + k_{y,i}\int(y_d - y)\right)\\ \tilde{\ddot{z}} &= \left(k_{z,d}(\dot{z_d} - \dot{z}) + k_{z,p}(z_d - z) + k_{z,i}\int(z_d - z)\right)\end{aligned} \quad (2.3.4)$$

где $k_{x,i}$, $k_{y,i}$, $k_{z,i}$ — постоянные вещественные числа, выбранные с учетом устойчивости положения равновесия замкнутой системы (1.2), (2.3.4).

Подобное изменение позволяет с течением времени привести летательный аппарат в желаемое положение. Для стабилизации в текущей точке пространства достаточно задать ее, как необходимую для достижения. Однако, интегральная составляющая данного регулятора порождает проблему перерегулирования (наглядный пример будет продемонстрирован в следующей главе).

Оптимизация полученного решения выполняется при помощи сохранения положительного воздействия интегральной составляющей регулятора (снижение отклонения от заданного положения с течением времени) и уменьшения ее недостатков. Рассмотрим работу алгоритма на примере управления движением по оси OX:

1. при изменении желаемого положения x_d интегральный коэффициент $k_{x,i}$ принимает значение 0, а вычисление интегральной суммы $\int(x_d - x)$ прекращается;
2. в момент достижения необходимого значения $x = x_d$ значение параметра $k_{x,i}$ меняется на исходное $k_{x,i}^{initial}$, а интегральная сумма начинает вычисляться с текущего момента времени t.

Подобный метод позволяет компенсировать воздействие внешних сил, действующих на квадрокоптер, с течением времени.

Глава 3. Реализация комплекса

3.1 Имитационное моделирование динамики движения летательного аппарата

Для реализации способов управления летательным аппаратом, приведенных ранее (§ 2.1), используются разработанные законы управления? представленные в § 2.3.

Процесс моделирования динамики движения квадрокоптера проводится на основе математической модели (1.2), рассмотренной выше, в среде MATLAB-Simulink. Схема Simulink-модели верхнего уровня приведена на рис. 6.

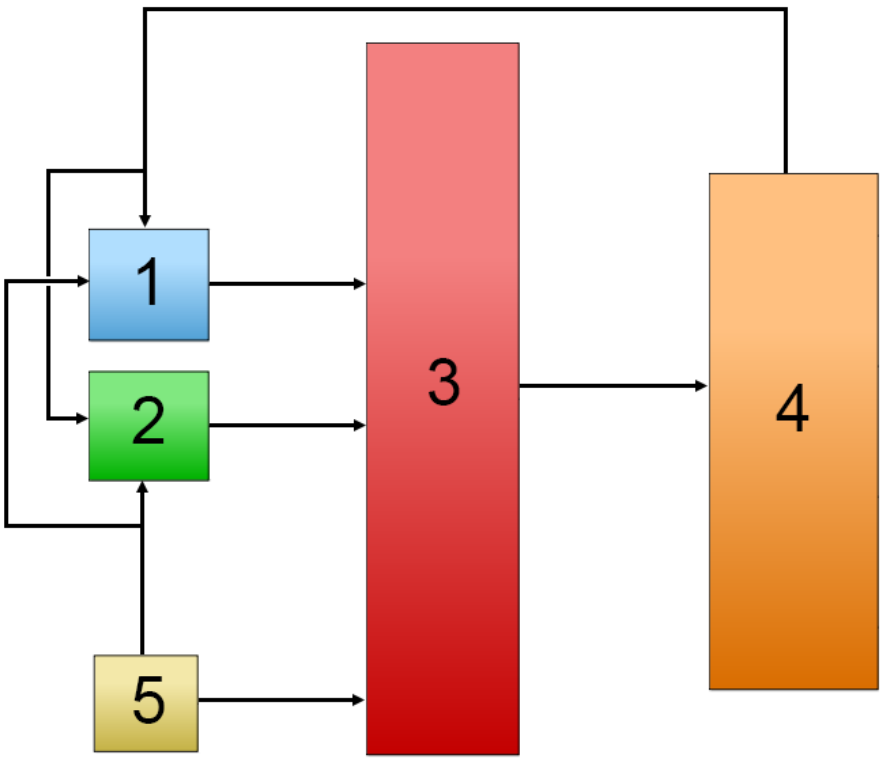


Рис. 6. Схема Simulink-модели

Блоки "1" и "2" реализуют алгоритмы управления (2.3.2), (2.3.3) и (2.3.4). В конкретный момент времени может работать только один из них

(зависит от текущего выбранного режима работы). Первый блок вычисляет управляющий сигнал для приведения квадрокоптера в определенную точку пространства (также содержит модификацию алгоритма: астатический регулятор по положению в пространстве). Второй блок использует закон управления для стабилизации по углам наклона для вычисления выходного сигнала. В третьем блоке содержится система дифференциальных уравнений, описывающая динамику летательного аппарата. Четвертый блок осуществляет вычисление измеряемой квадрокоптером информации. Пятый блок содержит константы, используемые в моделировании (табл. 1):

Параметр:	Значение:
m	0.5
g	9.81
I_x	$4.856 * 10^{-3}$
I_y	$4.856 * 10^{-3}$
I_z	$8.801 * 10^{-3}$
A_x	0.25
A_y	0.25
A_z	0.25
K	$3 * 10^{-6}$
L	0.25
B	$1.15 * 10^{-7}$

Таблица 1. Параметры компьютерной модели.

Время дискретизации компьютерного моделирования $T_{sample} = 0.005$с. Внешние воздействия на квадрокоптер задаются непосредственно в уравнениях динамики. Как видно из рис. 7 и 8, квадрокоптер является динамически неустойчивой системой при наличии даже незначительных внешних сил при отсутствии управления. Графики были получены при

моделировании движения квадрокоптера с постоянными оборотами моторов, заданными так, чтобы при отсутствии внешних возмущений компенсировать силу притяжения.

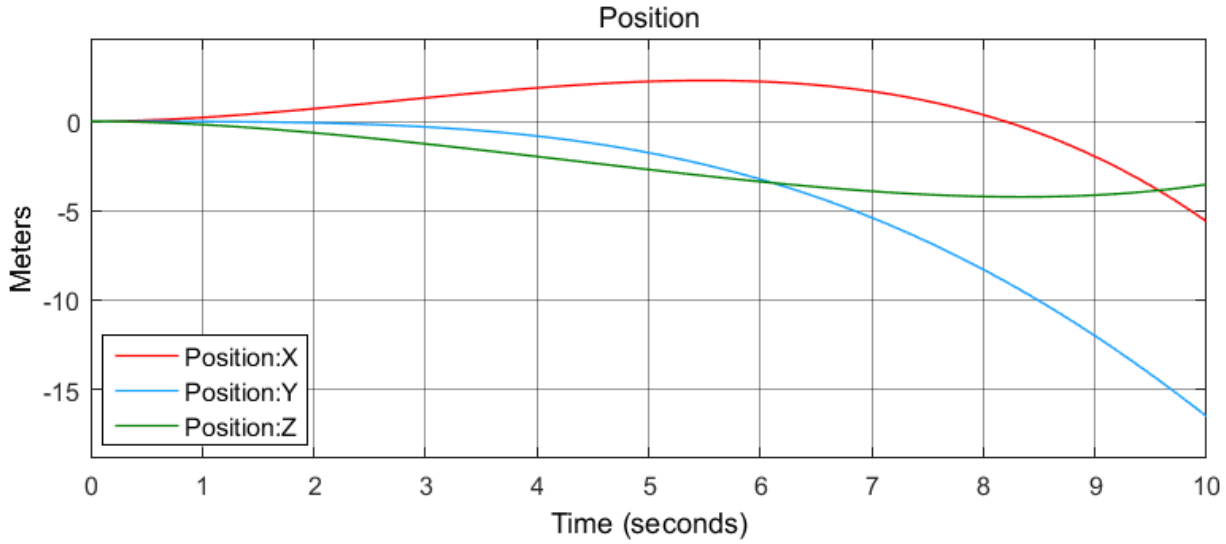


Рис. 7. Динамика движения летательного аппарата без закона управления (положение центра масс).

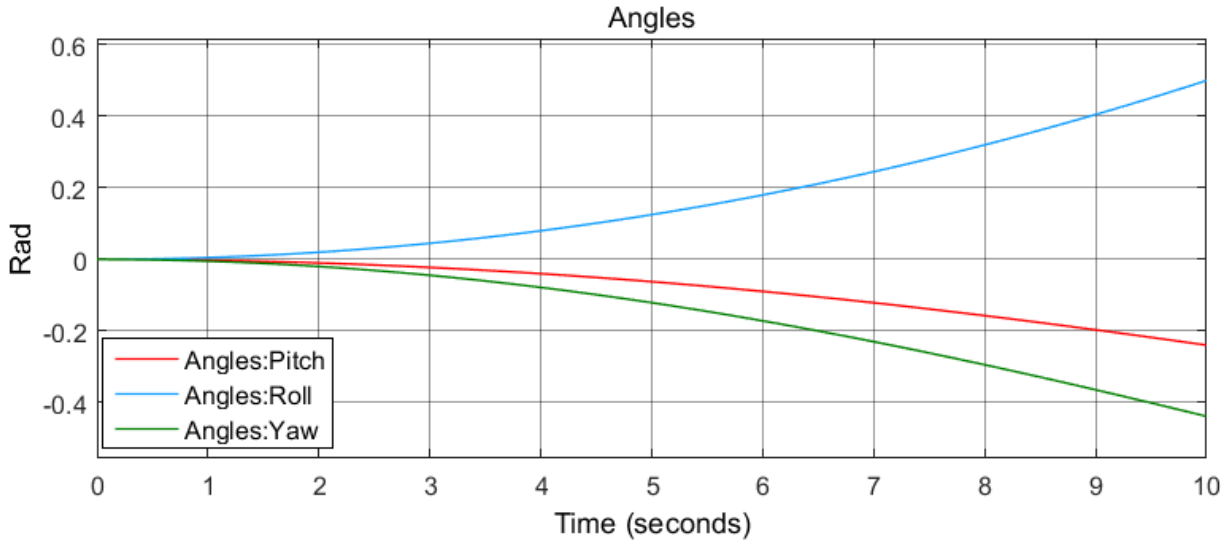


Рис. 8. Динамика движения летательного аппарата без закона управления (углы ориентации).

Далее рассмотрим применение к данной модели регулятора для стабилизации по углам наклона (рис. 9) (2.3.2). В качестве желаемых значений углов приняты следующие: тангаж $\varphi_d = 0$, крен $\theta_d = 0$, рысканье $\psi_d = 0$.

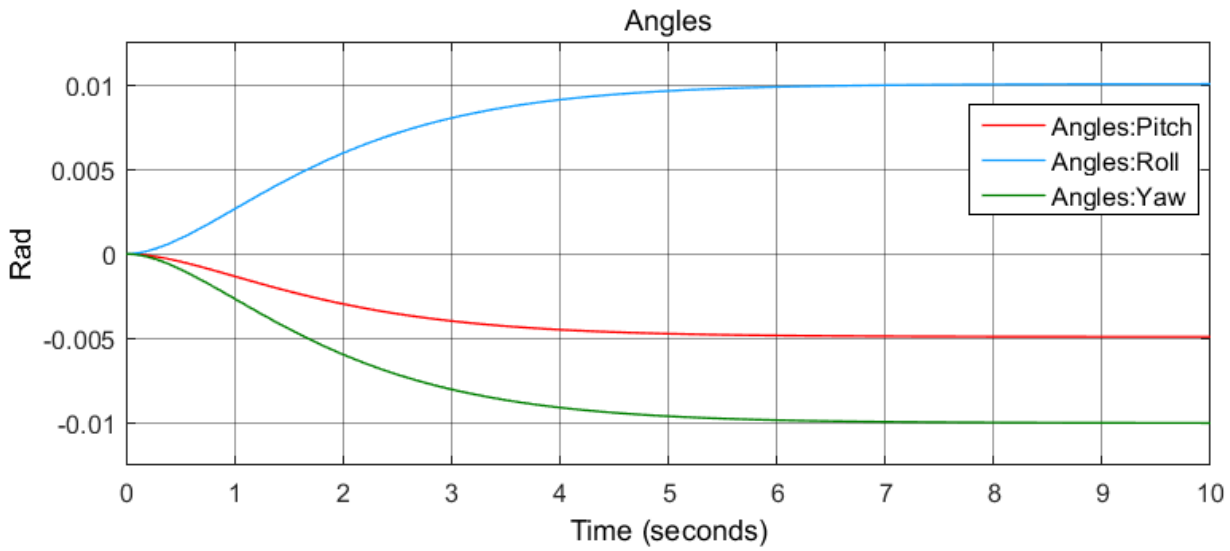


Рис. 9. Углы ориентации летательного аппарата (регулятор по углам наклона).

На графике видно, что при наличии внешнего воздействия регулятор обеспечивает положение равновесия замкнутой системы, отличное от желаемого. Отсутствие интегральной составляющей в данном регуляторе сказывается на небольшом отклонении от необходимого положения. Однако, без возможности устранить накопление ошибки, которое возникает из-за добавления интегрального слагаемого, применение последнего остается нежелательным.

Далее рассмотрим графики, отражающие динамику движения летательного аппарата с применением закона управления для приведения квадрокоптера в определенную точку пространства (2.3.3) (рис. 10 и 11):

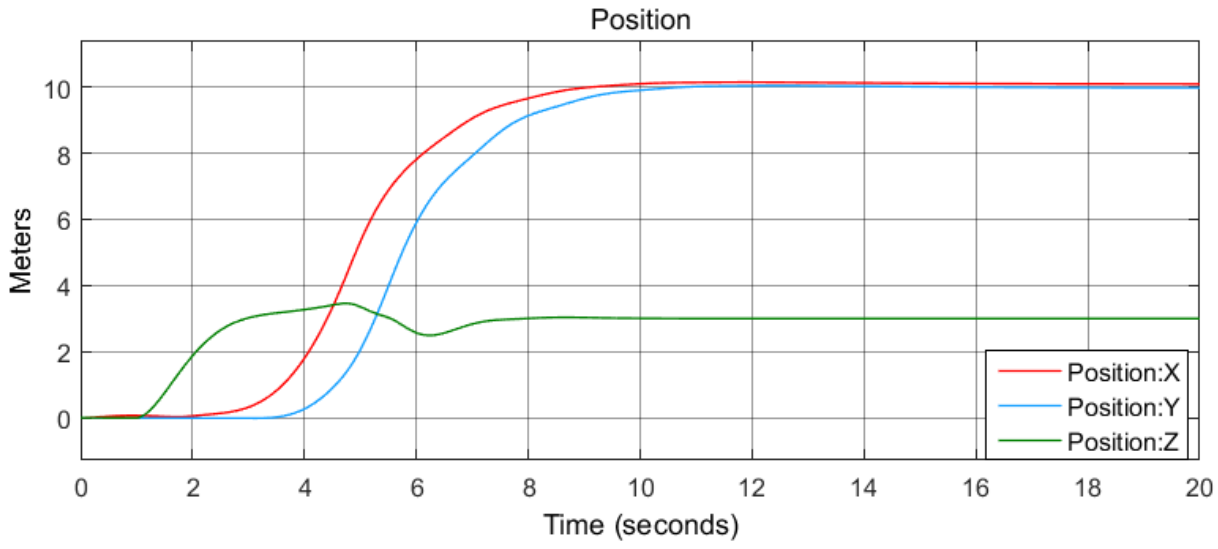


Рис. 10. Положение центра масс летательного аппарата с использованием регулятора для приведения в определенную точку пространства.

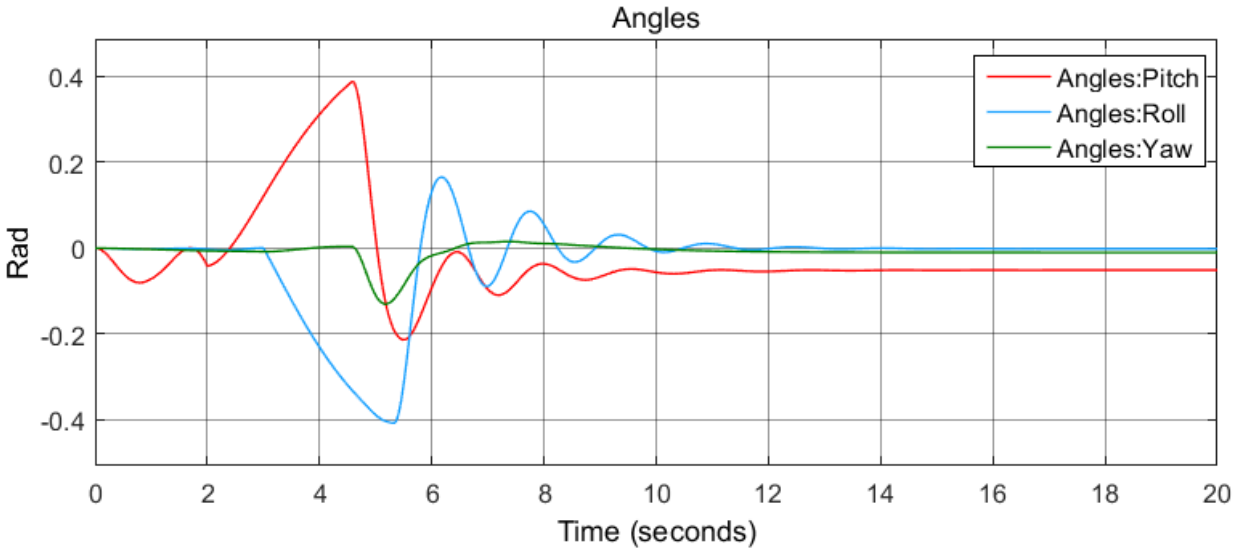


Рис. 11. Углы ориентации летательного аппарата с использованием регулятора для приведения в определенную точку пространства.

В качестве сигнала оператора выступает следующая последовательность команд: при времени симуляции $t = 1\text{с}$ задается координата $z_d = 3\text{м}$, при $t = 2\text{с}$ задается $x_d = 10\text{м}$, при $t = 3\text{с}$ задается $y_d = 10\text{м}$. В конечном итоге квадрокоптер должен стабилизироваться в точке $P_d(10, 10, 3)$. Согласно графику положения центра масс летательного аппарата (рис. 10), регулятор выполняет возложенную на него функцию, и весь переходный процесс с учетом хронологических сдвигов занимает около 9 секунд.

Для следующей симуляции увеличим воздействие внешнего возмущения. Пусть на квадрокоптер воздействует постоянный южный ветер с силой $F_e = 1.5\text{Н}$. Также в момент времени $t = 4\text{с}$ задается желаемый угол рысканья $\psi_d = 0.3\text{рад}$. Ниже приведены полученные графики (рис. 12 и 13):

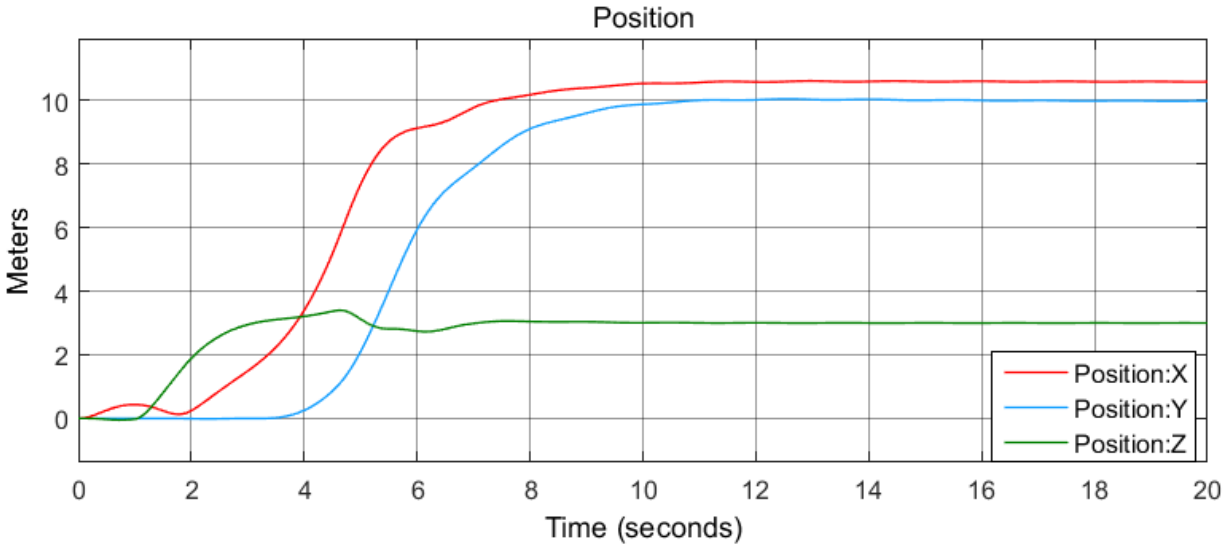


Рис. 12. Положение центра масс летательного аппарата с использованием регулятора для приведения в определенную точку пространства (увеличенное внешнее возмущение).

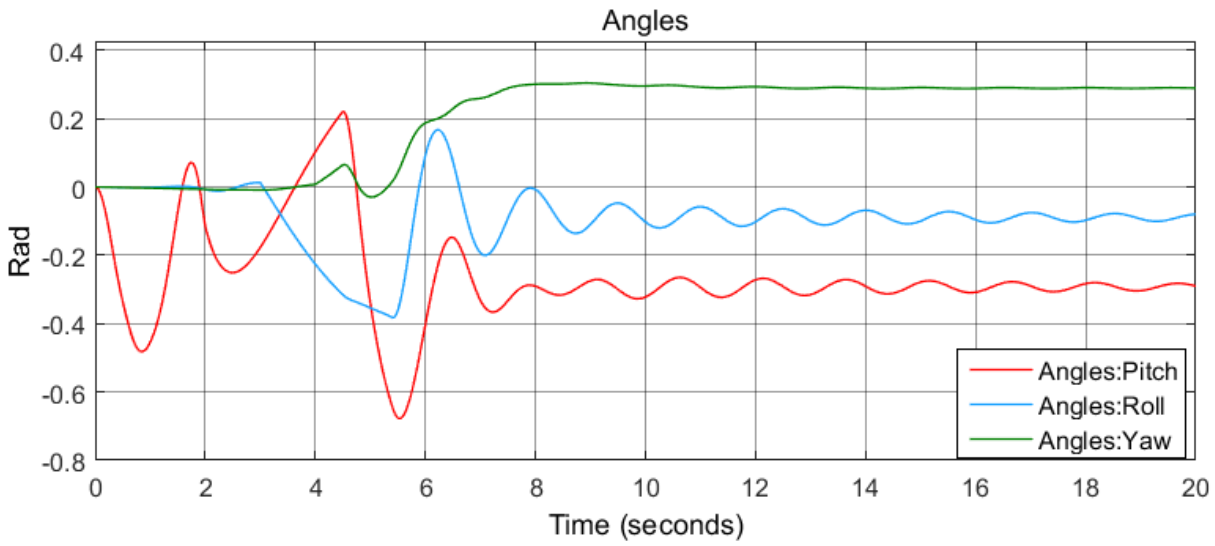


Рис. 13. Углы ориентации летательного аппарата с использованием регулятора для приведения в определенную точку пространства (увеличенное внешнее возмущение).

Регулятор осуществляет стабилизацию по заданному углу рысканья и выводит центр масс летательного аппарата на заданную позицию, но из-за внешнего воздействия возникает сильное отклонение по координате x: $x_{error} = 0.6$м.

Не изменяя начальных условий, рассмотрим применение астатического регулятора (2.3.4), являющегося модификацией предыдущего

алгоритма. На рис. 14 и 15 представлены графики динамики движения летательного аппарата:

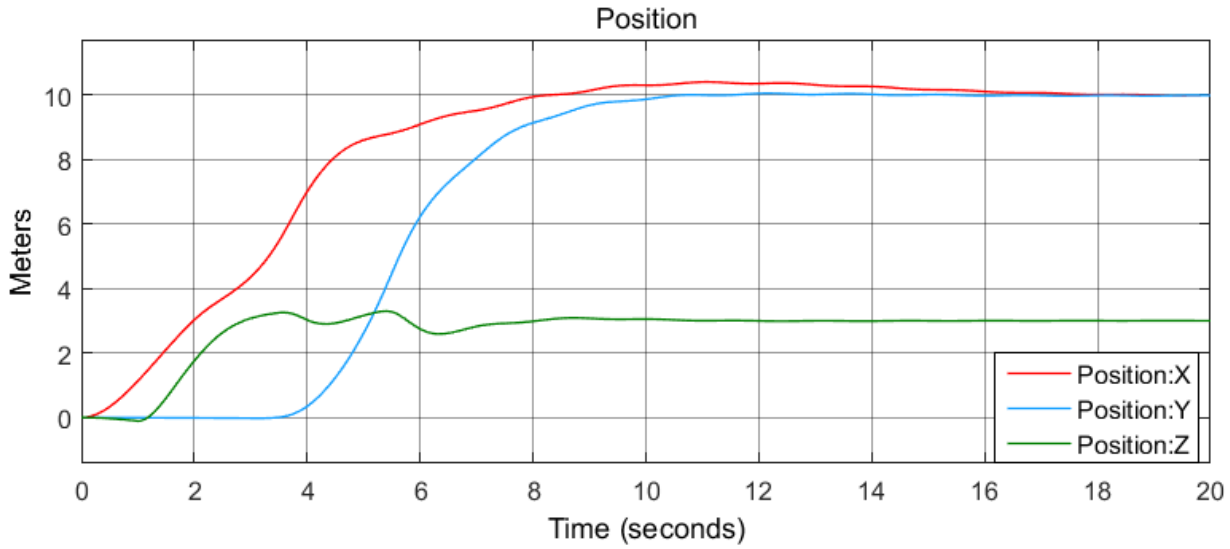


Рис. 14. Положение центра масс летательного аппарата с использованием астатического регулятора (увеличенное внешнее возмущение).

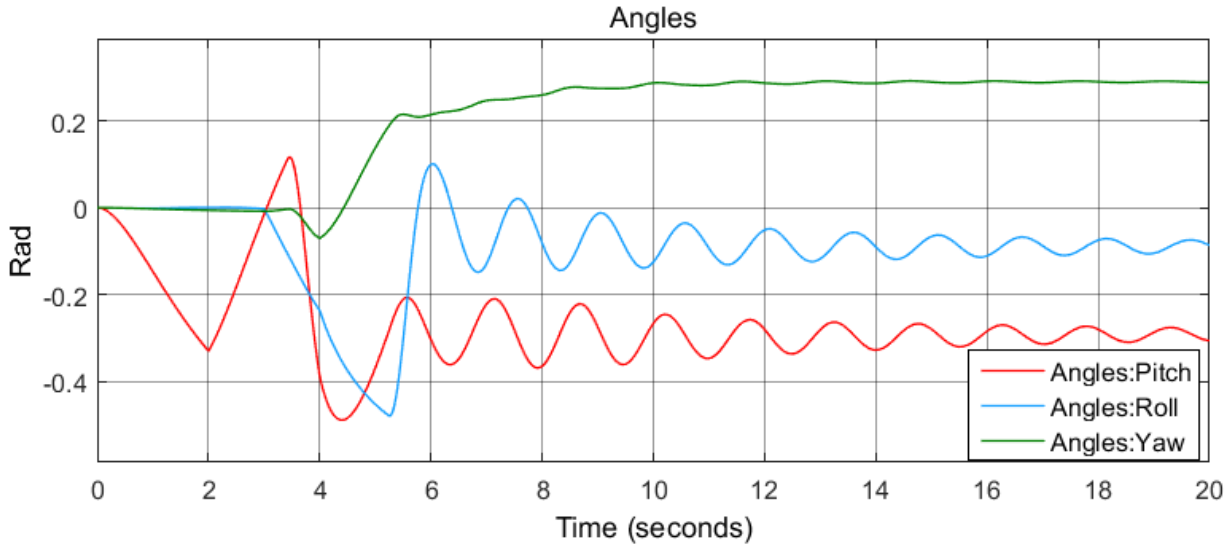


Рис. 15. Углы ориентации летательного аппарата с использованием астатического регулятора (увеличенное внешнее возмущение).

Разработанный закон управления позволяет компенсировать отклонение x_{error} с течением времени. Таким образом, для заданных параметров переходный процесс занимает около 15 секунд.

3.2 Практическая реализация комплекса

Базовая станция может быть воспроизведена на любом мобильном устройстве, удовлетворяющем требованиям, описанным в §2.1, с установленным программным приложением пульта управления. Окно управления с графическим интерфейсом пользователя показано на рис. 16:

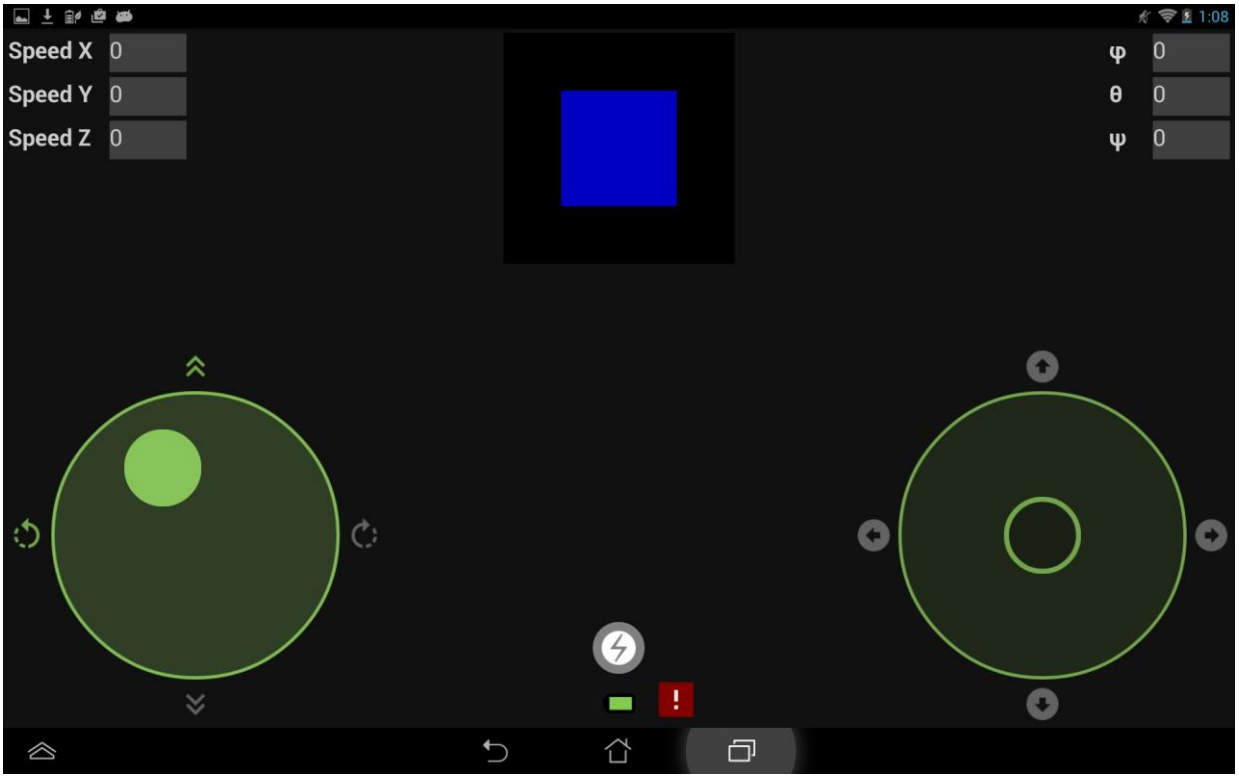


Рис. 16. Графический интерфейс окна управления летательным аппаратом.

Интерфейс содержит графическую реализацию двух рычагов управления. Вывод телеметрии осуществляется в текстовом виде, а также для этого используется 3d-модель, визуализирующая текущую ориентацию квадрокоптера (на рисунке представлена синей гранью трехмерной модели куба). Присутствуют кнопки дистанционного соединения с летательным аппаратом и аварийной остановки электромоторов. Переключение между режимами управления происходит при нажатии на индикатор уровня заряда батарей.

Отладка комплекса производится при помощи консоли оператора, показанной на рис. 17. Для удобства пользователя реализована система

справки, содержащая сведения о доступных командах.

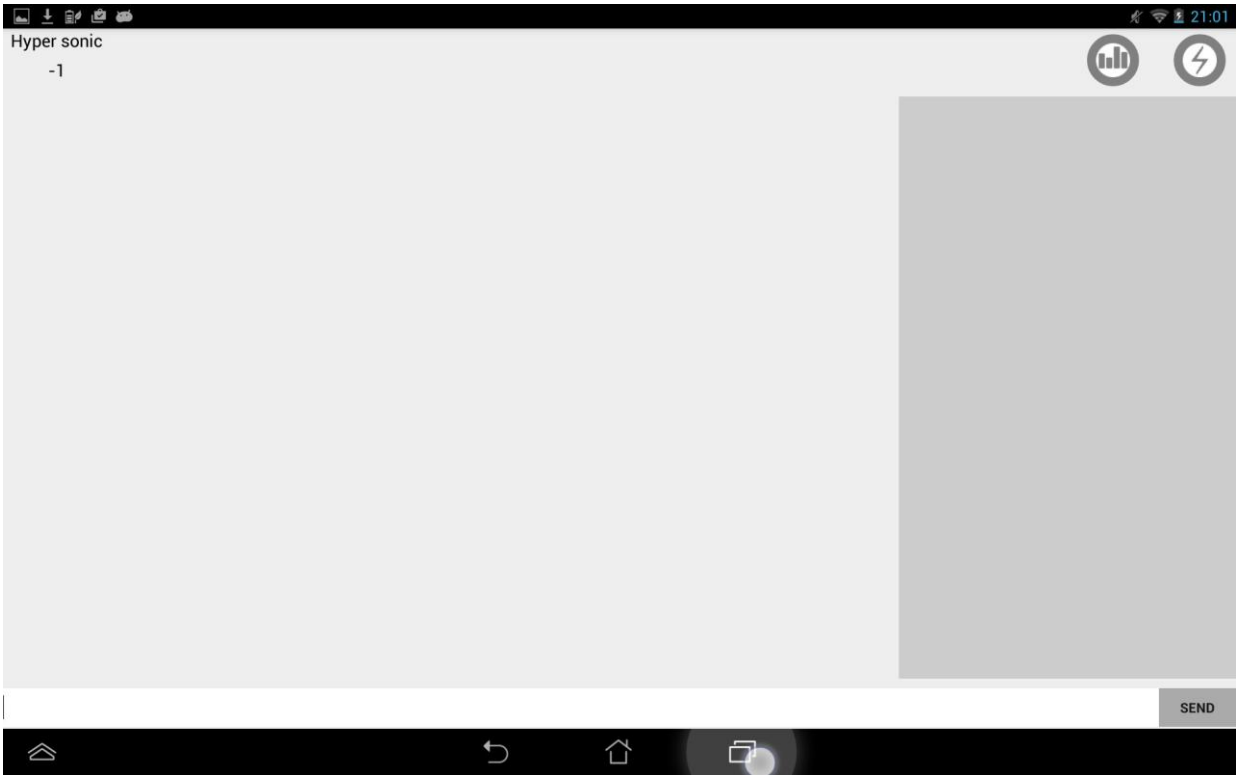


Рис. 17. Консоль оператора.

Сборка летательного аппарата производилась с учетом предъявляемых к комплексу требований. Основные компоненты бортовой части представлены в табл. 2.

Вычислительный блок	
Плата микроконтроллера	Stm32f407 Discovery
Блок коммуникации	
Беспроводной интерфейс	WIFI232-A
Блок исполнительных устройств	
Электромотор	4x NTM 2830-1100KV
Электронный регулятор хода	4x Turnigy Multistar 30 Amp Multi-rotor Brushless ESC 2-4S
Блок датчиков	
Инерциальное измерительное устройство	MPU9255

Инфракрасный датчик препятствий	LM393
Ультразвуковой датчик расстояния	HC-SR04
Датчик атмосферного давления (барометр)	BMP180

Таблица 2. Компоненты летательного аппарата.

Бортовое программное обеспечение реализовано на языке программирования C++ 11 и содержит основной функционал, описанный в параграфе 2.2.

В настоящий момент практическая реализация летательного аппарата находится на стадии предполетного тестирования.

Выводы

В ходе настоящей научно-исследовательской работы были получены следующие результаты, которые выносятся на защиту:

- Разработана логика и структура работы аппаратно-программного комплекса для дистанционного управления движением квадрокоптера.
- В составе комплекса разработаны бортовые алгоритмы управления для стабилизации летательного аппарата с определенными углами ориентации; для приведения летательного аппарата в заданную точку пространства с заданным углом рысканья; для компенсации внешнего воздействия на положение центра масс летательного аппарата в пространстве. Произведено имитационное моделирование динамики движения летательного аппарата с применением разработанных законов управления.

К преимуществам комплекса стоит отнести применение в управлении астатического регулятора и реализацию аварийных алгоритмов. Высокая производительность бортовой аппаратной части позволяет использовать данное решение в дальнейших исследованиях с применением фото и видео устройств.

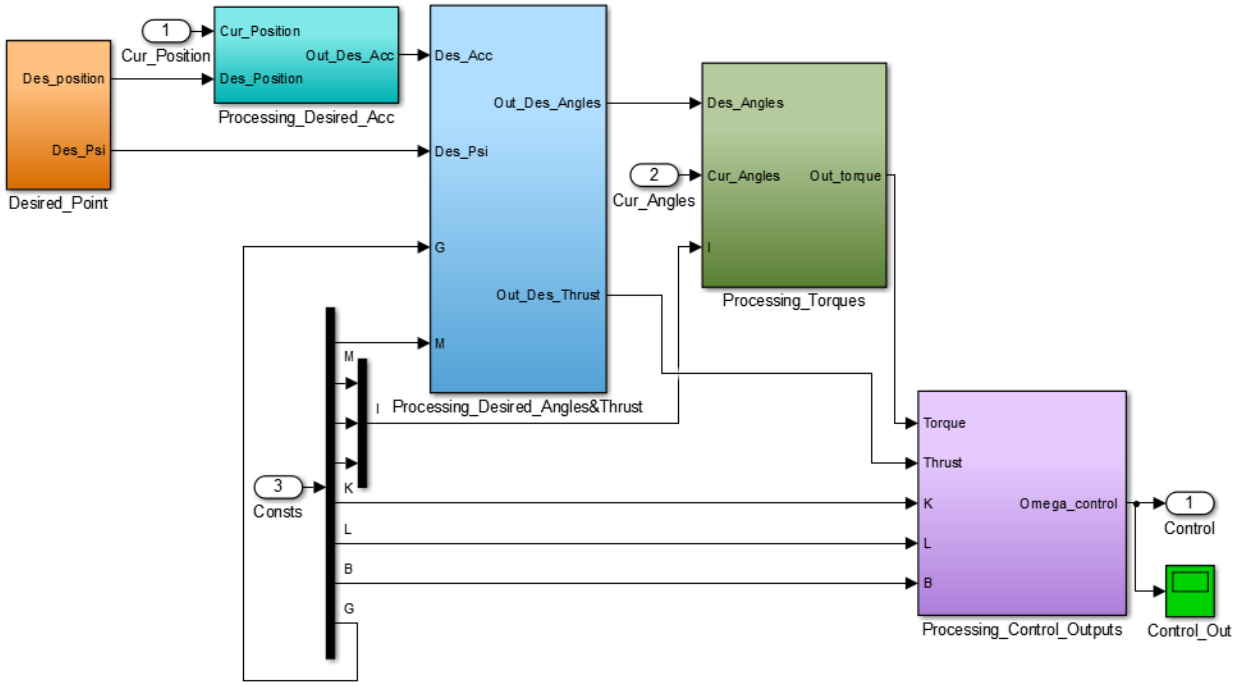
Планируется возможность применения других стандартов беспроводной связи для увеличения доступного радиуса удаления бортовой части от базовой станции, а также использование датчиков геолокации для задачи автоматизированного полета квадрокоптера вне зоны обнаружения базовой станции.

Список литературы

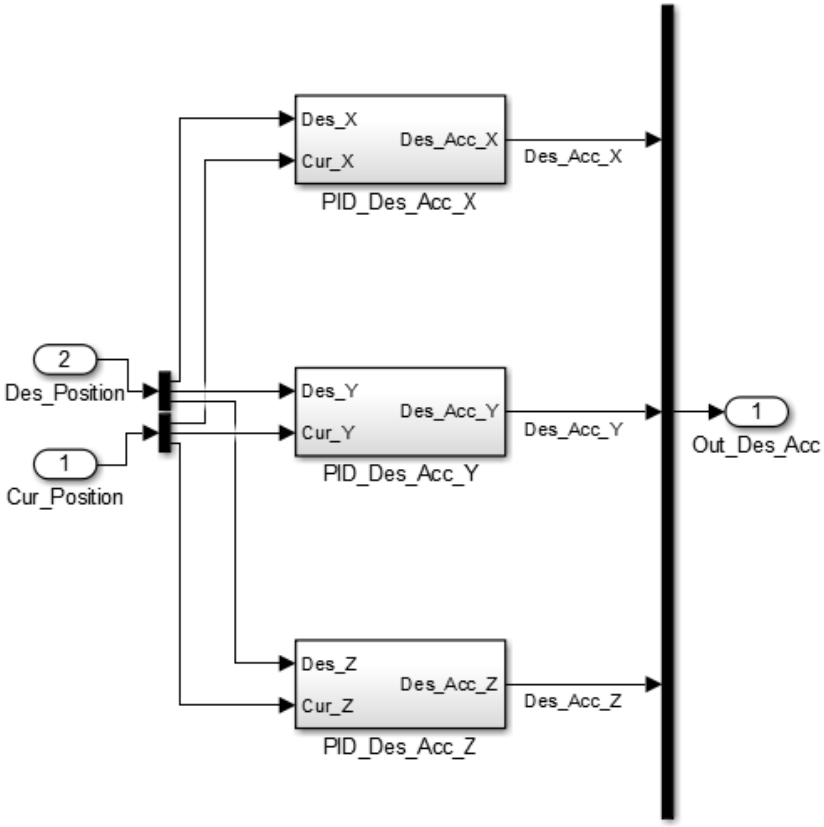
1. Balas C. Modelling and linear control of a quadrotor. MSc thesis. Cranfield University, 2007. 150 c.
2. ArduPilot. http://ardupilot.org/ardupilot/index.html
3. DJI Phantom 4. http://www.dji.com/product/phantom-4
4. Teppo Luukkonen. Modelling and control of quadcopter. Espoo, 2011. 26 c.
5. D. Mellinger. Trajectory Generation and Control for Quadrotors. University of Pennsylvania, 2012. 137 c.
6. Датчик давления BMP180 (BMP085). http://robotclass.ru/tutorials/arduino-pressure-sensor-bmp180-bmp085
7. Speed of sound. https://en.wikipedia.org/wiki/Speed_of_sound
8. R. Mahony, Sung Han Cha, T. Hamel. A coupled estimation and control analysis for attitude stabilization of mini aerial vehicles // Australasian Conference on Robotics and Automation. Australian National University 2006. 10 c.

Приложение

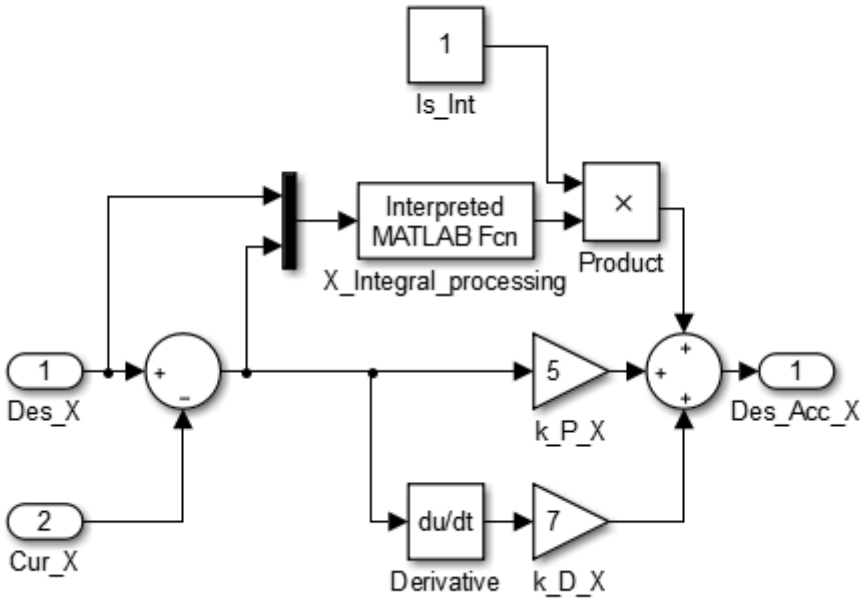
Блок реализации алгоритма управления для приведения квадрокоптера в точку:



Блок вычисления желаемых ускорений (верхний уровень):



Блок вычисления желаемых ускорений (нижний уровень):



Код блока "X_Integral_Processing":

```
function [ value ] = X_Integral_coef_processing( EMP )
    global param_value_X_Integral;
    global last_des_X;
    global int_sum_X;
    Des_X = EMP(1);
    Cur_Err = EMP(2);
    if last_des_X ~= Des_X
        last_des_X = Des_X;
        param_value_X_Integral = sign(Cur_Err);
        coef = 0;
    else
        if param_value_X_Integral ~= 0
            if param_value_X_Integral ~= sign(Cur_Err)
                coef = 0.5;
                param_value_X_Integral = 0;
                int_sum_X = 0;
            else
                coef = 0;
            end
        else
            coef = 0.5;
        end
    end
    int_sum_X = int_sum_X + Cur_Err * 0.005;
    value = coef * int_sum_X;
end
```

Пример алгоритма интерпретации команды на стороне базовой станции:

```
String parse(String input){
    String temp = "";
    Vector<String> vector_input;
    if (input.contains("\n")) {
        String[] rows = input.split("\n");
```

```
        for (String i : rows) {
            vector_input = new Vector<>();
            if (i.contains(" ")) {
                String[] parts = i.split(" ");
                Collections.addAll(vector_input, parts);
            } else vector_input.add(i);
            if (vector_input.size() != 0) {
                if (vector_input.get(0).equals("help")) {
                    if (hmRunnable_table.containsKey(vector_input.get(1)))
                        temp += !temp.equals("") ? "\n" +
hmRunnable_table.get(vector_input.get(1)).run(vector_input) :

hmRunnable_table.get(vector_input.get(1)).run(vector_input);
                    else temp += !temp.equals("") ? "\n" + "Unknown command:
" + vector_input.get(1) :
                                "Unknown command: " + vector_input.get(1);
                } else {
                    if (hmRunnable_table.containsKey(vector_input.get(0)))
                        temp += !temp.equals("") ? "\n" +
hmRunnable_table.get(vector_input.get(0)).run(vector_input) :

hmRunnable_table.get(vector_input.get(0)).run(vector_input);
                    else temp += !temp.equals("") ? "\n" + "Unknown command:
" + vector_input.get(0) :
                                "Unknown command: " + vector_input.get(0);
                }
            }
        }
    } else {
        vector_input = new Vector<>();
        if (input.contains(" ")) {
            String[] parts = input.split(" ");
            Collections.addAll(vector_input, parts);
        } else {
            vector_input.add(input);
        }
        if (vector_input.size() != 0) {
            if (vector_input.get(0).equals("help")) {
                if (hmRunnable_table.containsKey(vector_input.get(1)))
                    temp += !temp.equals("") ? "\n" +
hmRunnable_table.get(vector_input.get(1)).run(vector_input) :

hmRunnable_table.get(vector_input.get(1)).run(vector_input);
                else temp += !temp.equals("") ? "\n" + "Unknown command: " +
vector_input.get(1) :
                            "Unknown command: " + vector_input.get(1);
            } else {
                if (hmRunnable_table.containsKey(vector_input.get(0)))
                    temp += !temp.equals("") ? "\n" +
hmRunnable_table.get(vector_input.get(0)).run(vector_input) :

hmRunnable_table.get(vector_input.get(0)).run(vector_input);
                else temp += !temp.equals("") ? "\n" + "Unknown command: " +
vector_input.get(0) :
                            "Unknown command: " + vector_input.get(0);
            }
        }
    }
    return temp;
}
```